

CLAIMS

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A method for implementing a polymorphic call site in a computer system executing an object oriented program, said method comprising the steps of:

a) creating a template of a polymorphic inline cache for a polymorphic call site, the template having a plurality of slots;

b) executing the polymorphic inline cache with an object of type k;

c) invoking a polymorphic inline cache initialisation routine;

d) finding an available k slot of the polymorphic inline cache;

e) locking the k slot of the polymorphic inline cache;

f) searching for a k method to call for the object of type k;

g) filling the k slot with a call instruction to the k method;

h) unlocking the k slot to complete the k slot; and

i) calling the k method of the object of type k.

2. The method of Claim 1, wherein a first thread may initialize and/or access the k slot at the same time a second thread is initializing and/or accessing a slot other than the k slot.

3. The method of claim 2, wherein a second thread calling the k method while the first thread is initializing the k slot waits until after said unlocking step (h).



3 4. The method of Claim 2, wherein a second thread calling the k method while the first
4 thread is initializing the k slot searches for and calls the k method and leaves the
5 polymorphic inline cache unchanged.

1 5. The method of Claim 1, wherein a polymorphic inline cache is created for each
2 polymorphic call site.

1 6. The method of Claim 1, wherein said creating step (a) further comprises inserting a first
2 illegal type value in a compare instruction of every slot of the polymorphic inline cache to
3 indicate each slot is empty.

1 7. The method of Claim 1, wherein said creating step (a) site further comprises putting a
2 bit in each slot to indicate that each slot is empty.

1 8. The method of Claim 6, wherein said locking step (e) further comprises replacing the first
2 illegal type value in the compare instruction of the k slot with a second illegal type value to
3 indicate the k slot is in use.

1 9. The method of Claim 1, wherein said locking step (e) further comprises changing a bit
2 in the k slot to indicate that the k slot is in use.

1 10. The method of Claim 8, wherein said unlocking step (h) further comprises replacing the
2 second illegal type value in the compare instruction of the k slot with a value of type k.

1 11. The method of Claim 1, further comprising the step of (j) updating the polymorphic inline
2 cache so that an object of type k+1 will initialise a corresponding k+1 slot.

1 12. The method of Claim 11, wherein said updating step (j) further comprises inspecting the
2 polymorphic inline cache to find the next empty slot.

1 13. The method of Claim 12, wherein said polymorphic inline cache initialisation routine is
2 the same for every object of the polymorphic inline cache.



1 14. The method of Claim 11, wherein said updating step (j) further comprises maintaining
2 a state of the k slot or the k+1 slot.

1 15. The method of Claim 14, wherein said polymorphic inline cache initialization routine is the
2 same for every object of the polymorphic inline cache.

1 16. The method of Claim 15, wherein said updating step (j) further comprises modifying the
state of the k+1 slot.

1 17. The method of Claim 1, wherein said invoking step (c) further comprises calling a
2 different initialization routine for every object of a different type.

1 18. The method of Claim 11, wherein said updating step (j) further comprises modifying the
2 initialization routine so that upon a cache miss of the k slot, the k+1 initialization routine is
3 called.

1 19. A system in a computing environment for implementing a polymorphic call site,
2 comprising:

3
4 a central processing unit for executing an application;

5
6 memory connected to the central processing unit via a bus;

7
8 at least one input/output device connected to the bus and connected to a network
9 interface to an external computer network;

10
11 a just-in-time compiler for compiling object oriented applications for execution; and

12 a polymorphic inline cache created by said just-in-time compiler, said polymorphic
13 inline cache implementing a lockable slot for each individual object type to a
14 polymorphic call site in the application.

1 20. A system in a computing environment for executing an application, comprising:



- 2 a processor for executing the application;
- 3 memory connected to the processor with an internal bus;
- 4 a network interface to connect to an external computer network; and
- 5 a polymorphic inline cache having a plurality of slots, each slot allocated to an
- 6 object type of a method and locked to other object types of a polymorphic call site in the application.

- 1 21. A method of using a polymorphic inline cache, the method comprising the steps of:
- 2 calling a first method having a first object type from an executing object oriented
- 3 program;
- 4 locking a first slot of the polymorphic inline cache with a call to the first method of
- 5 the first object type; and
- 6 executing the first slot of the polymorphic inline cache.

- 1 22. The method of Claim 21, further comprising the steps of:
- 2 calling a second method having a second object type;
- 3
- 4 locking a second slot of the polymorphic inline cache with a call to the second
- 5 method of the second object type; and
- 6 executing the second slot of the polymorphic inline cache.

- 1 23. The method of Claim 22, wherein the first slot of the polymorphic inline cache is executed
- 2 simultaneously with the second slot of the polymorphic inline cache.



1 24. The method of Claim 22, wherein the first method of the first object type is called from
2 a first thread which executes independently from an executing second thread which called
3 the second method of the second object type.

1 25. The method of Claim 21, further comprising the steps of:

2 calling a second method having a second object type;

3 waiting until the first method of the first object type has executed;

4 determining whether the second object type and the first object type are the same
5 object type; and

6 not locking a second slot of the polymorphic inline cache with a call to the method
7 of the second type if the first and second object types are the same object type; and

8 locking the second slot with a call to the second method if the first and second
9 object types are not the same object type and executing the second slot.

1 26. A computer system for executing an object oriented program, comprising:

2 means for executing an object oriented program;

3 means for calling a first method from a first slot of a polymorphic inline cache;

4 means for calling a second method from a second slot of the polymorphic inline
5 cache;

6 means for determining if the first method and the second method have an identical
7 object type;

8 means for calling the first method and the second method simultaneously if they do
9 not have the identical object type; and

10 means for preventing calling the second method from the second slot of the
11 polymorphic inline cache until said means for calling the first method has completed
12 if they have the identical object type.

13 27. The computer system of Claim 26, wherein the first method may be called from a first
14 thread and the second method may be called from an independently executing second
15 thread.

1 28. The computer system of Claim 27, further comprising:

2 means for invoking a first polymorphic inline cache initialisation routine;

3 means for locking the first slot of the polymorphic inline cache;

4 means for filling the first slot with a call instruction to the first method while the first
5 slot is locked;
6



7 means for updating the polymorphic inline cache so that a second method of a
8 second type will invoke a second polymorphic inline cache initialisation routine;

9 means for making the first slot available to the first method;

10 means for calling the first method;

11 means for locking the second slot of the polymorphic inline cache;

12 means for filling the second slot with a call instruction to the second method while
13 the second slot is locked; and

14 means for updating the polymorphic inline cache so that a Nth method of a Nth type
15 will invoke a Nth polymorphic inline cache initialization routine.

16
17
18
19
20
21
22
23
24
25
26
27
28
29. The method of Claim 28, wherein the first, second, and Nth polymorphic inline cache
initialization routines are identical.

30. The method of Claim 28, wherein the first polymorphic inline cache initialisation routine
is called from a first thread simultaneously while an independently executing second or Nth
thread is calling the second or Nth method or is calling the second or Nth polymorphic inline
cache initialization routine.